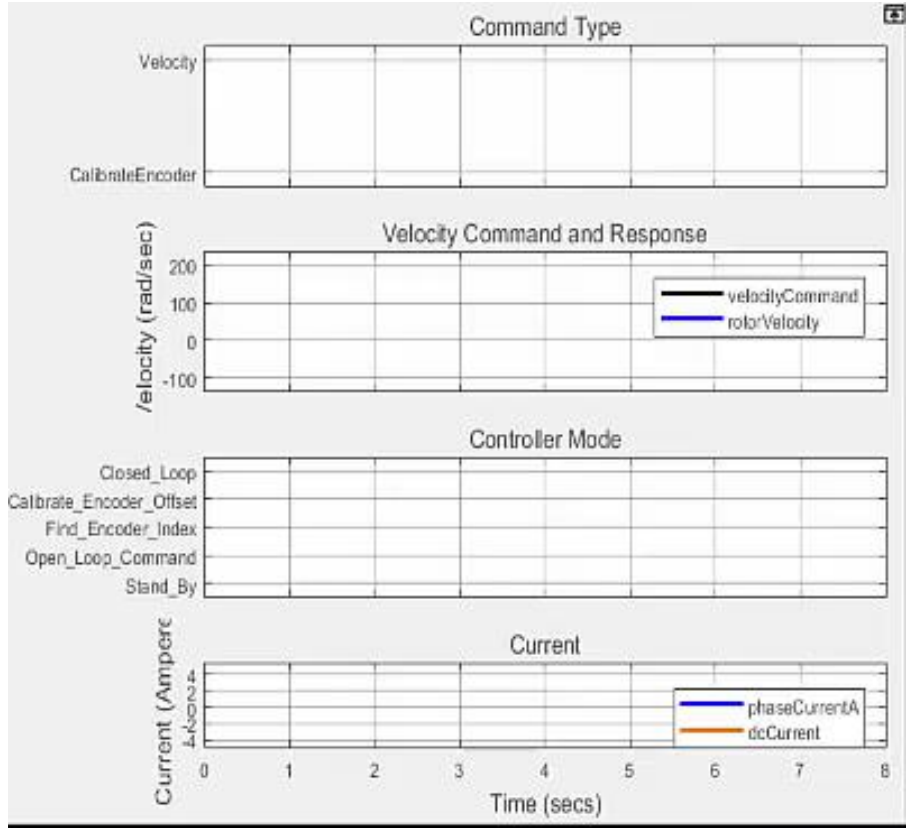
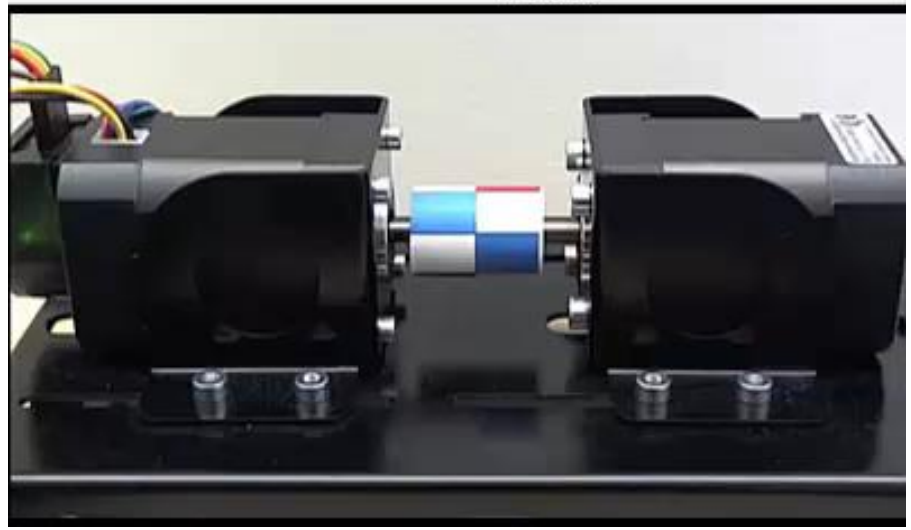
MATLAB EXPO 2018

Co-diseño Hardware-Software para Control de Motores

Luis López


Command Type
Velocity
CalibrateEncoder
Velocity Command and Response
/elocity (rad/sec)
200
100
0
-100
velocityCommand
rotorVelocity
Controller Mode
Closed_Loop
Calibrate_Encoder_Offset
Find_Encoder_Index
Open_Loop_Command
Stand_By
Current
Current (Amperc
phaseCurrentA
dcCurrent
0
1
2
3
4
5
6
7
8
Time (secs)



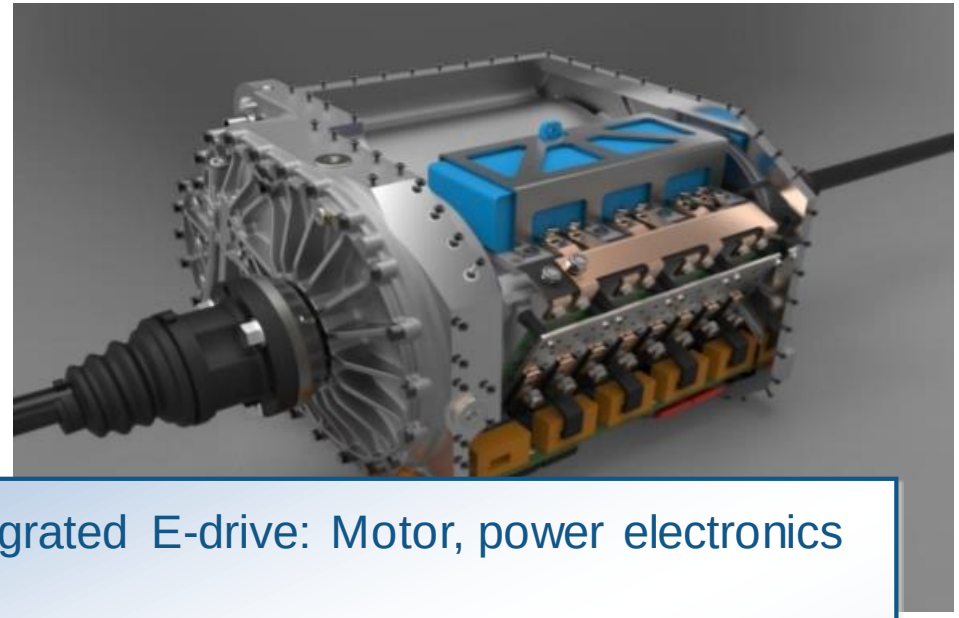
Takeaways

Model-Based Design for SoC FPGAs

- Enables early validation of specifications using simulation
- Improves design team collaboration and designer productivity.
- Reduces hardware testing time by 5x

Punch Powertrain develops complex SoC-based motor control

- Powertrains for hybrid and electric vehicles
- Need to increase power density and efficiency at a reduced cost
  - Integrate motor and power electronics in the transmission
- New switched reluctance motor
  - Fast: 2x the speed of their previous motor
    - Target to a Xilinx® Zynq® SoC 7045 device
  - Complex: 4 different control strategies
- No experience designing FPGAs!



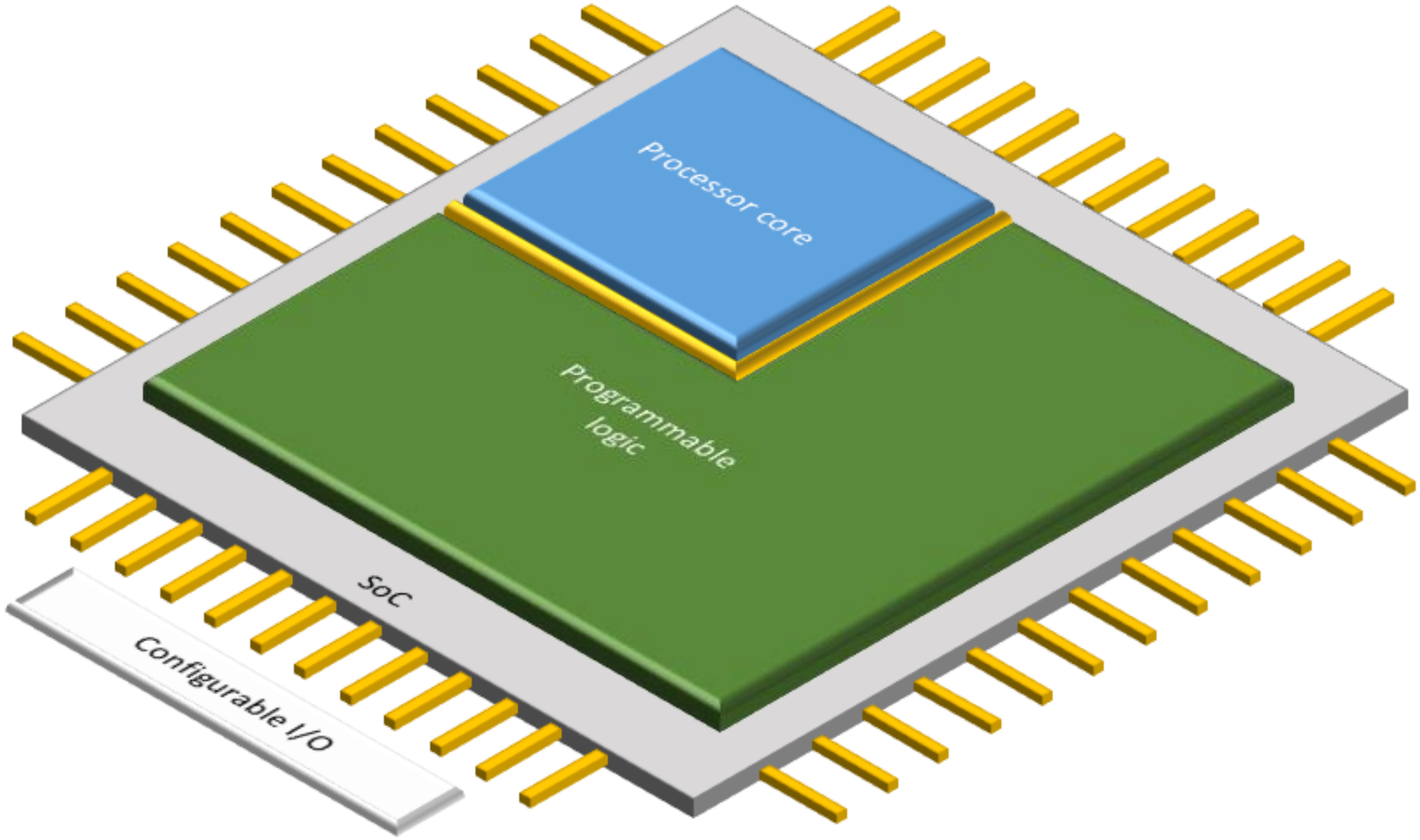
- ✓ Designed integrated E-drive: Motor, power electronics and software
- ✓ 4 different control strategies implemented
- ✓ Completed in 1.5 years with 2FTE's
- ✓ Models reusable for production
- ✓ Smooth integration and validation due to development process – thorough validation before electronics are produced and put in the testbench

Link to video of presentation

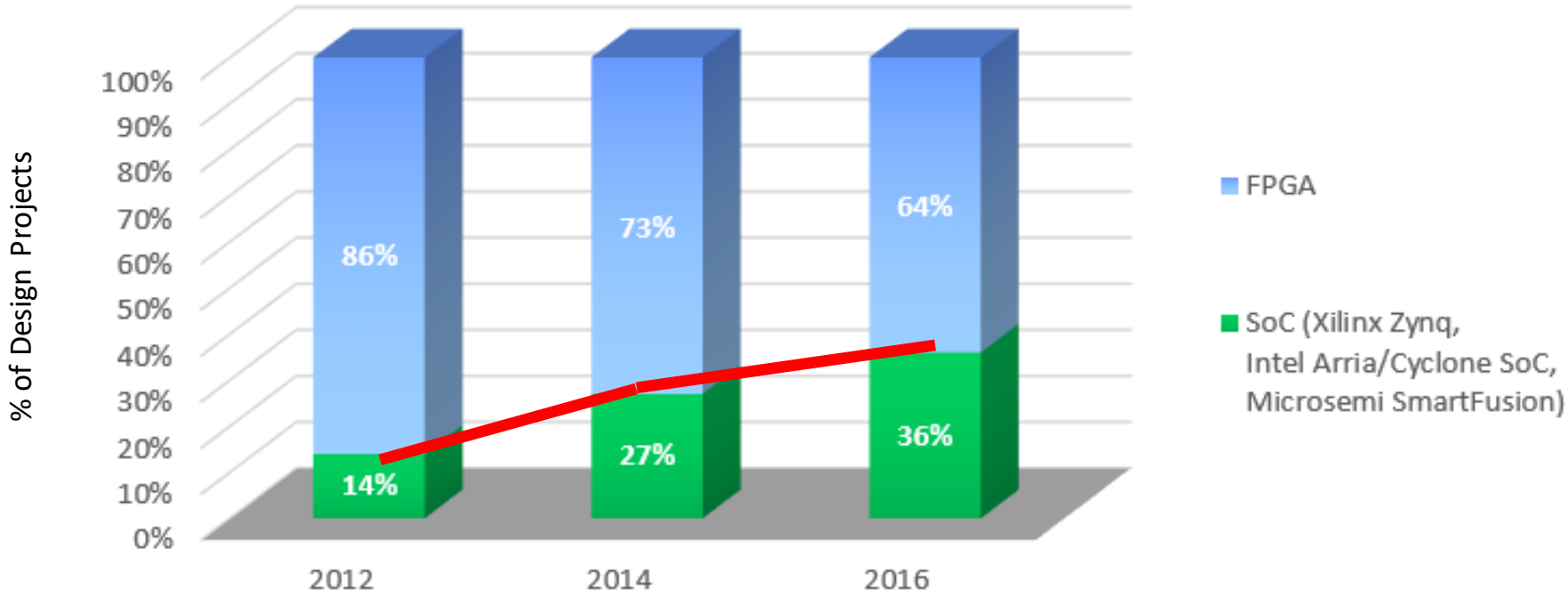
Key trend: Increasing demands from motor drives

- Advanced algorithms require faster computing performance.
  - Field-Oriented Control
  - Sensorless motor control
  - Vibration detection and suppression
  - Multi-axis control

What's an SoC?



Key Trend: SoCs are now used in 36% of new FPGA projects



Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Challenges in using SoCs for Motor and Power Control

- Integration of software and hardware partitions of algorithm on SoC drives need for collaboration
- Validation of design specifications with limits on access to motors in labs.
- How to make design decisions that cut across system components?

Why use Model-Based Design to develop motor control applications on SoCs?

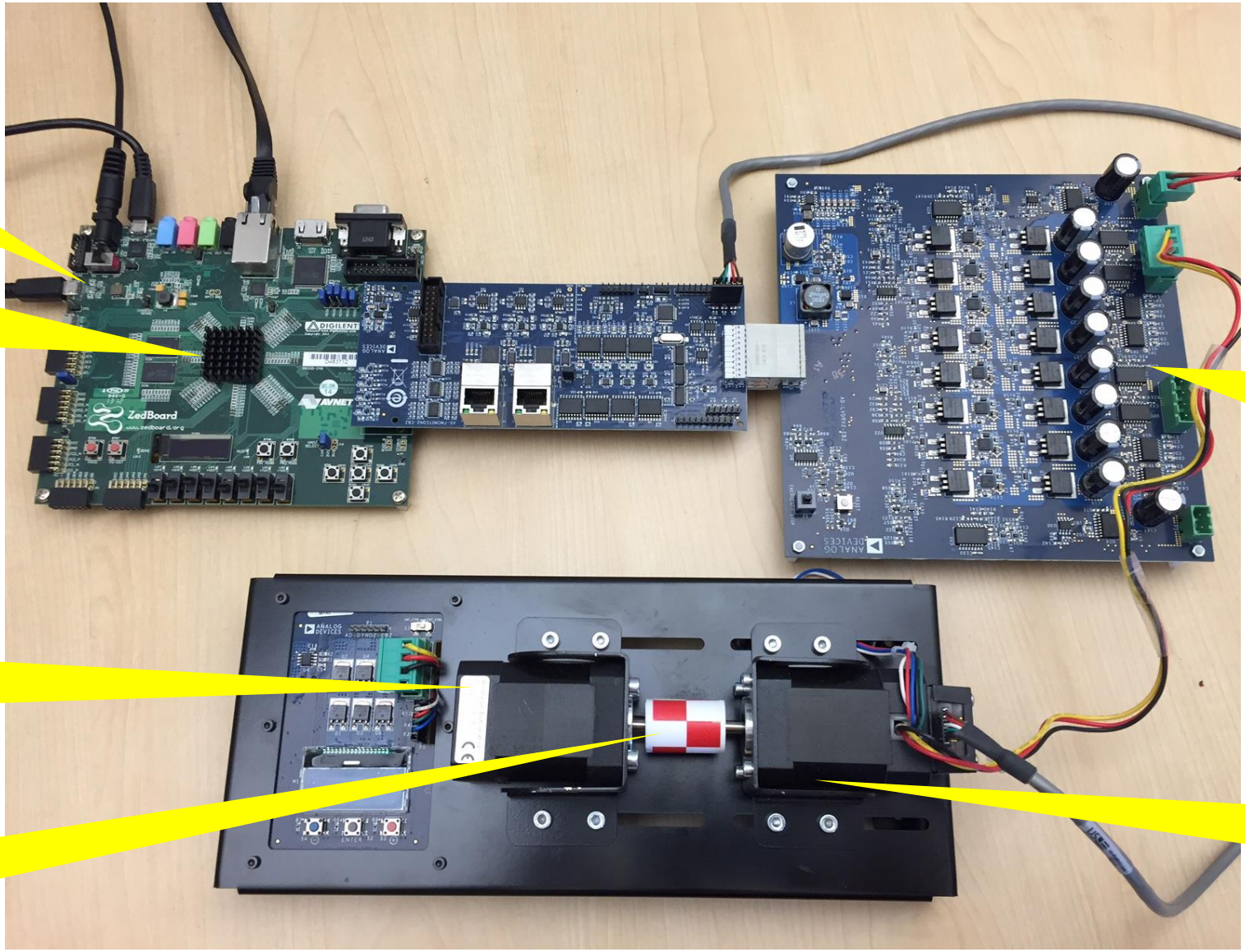
- Enables early validation of specifications using simulation months before hardware is available.
- Improves design team collaboration and designer productivity by using a shared design environment.
- Reduces hardware testing time by 5x by shifting design from lab to the desktop

ZedBoard

Zynq SoC
(XC7Z020)

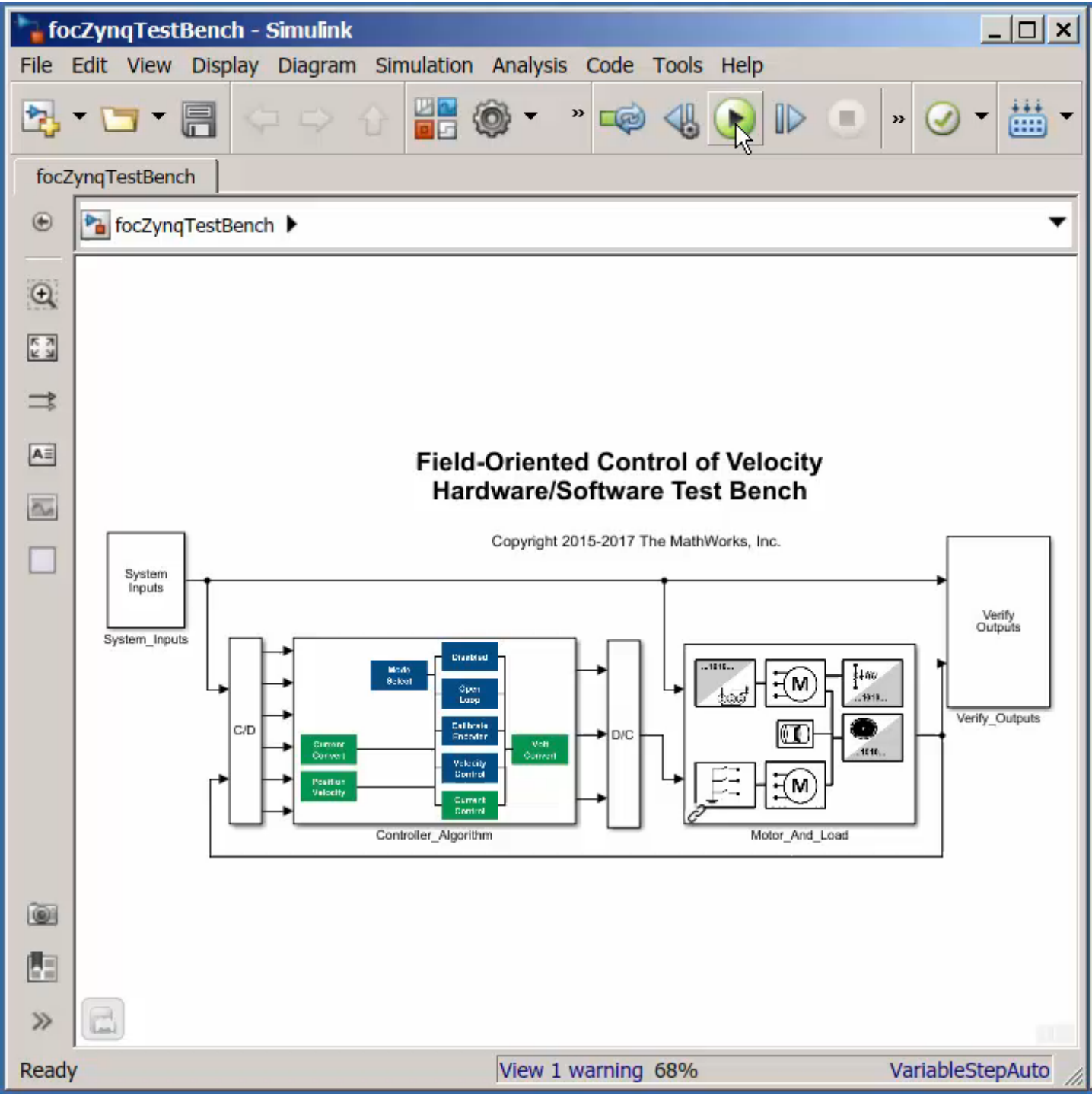
Load motor

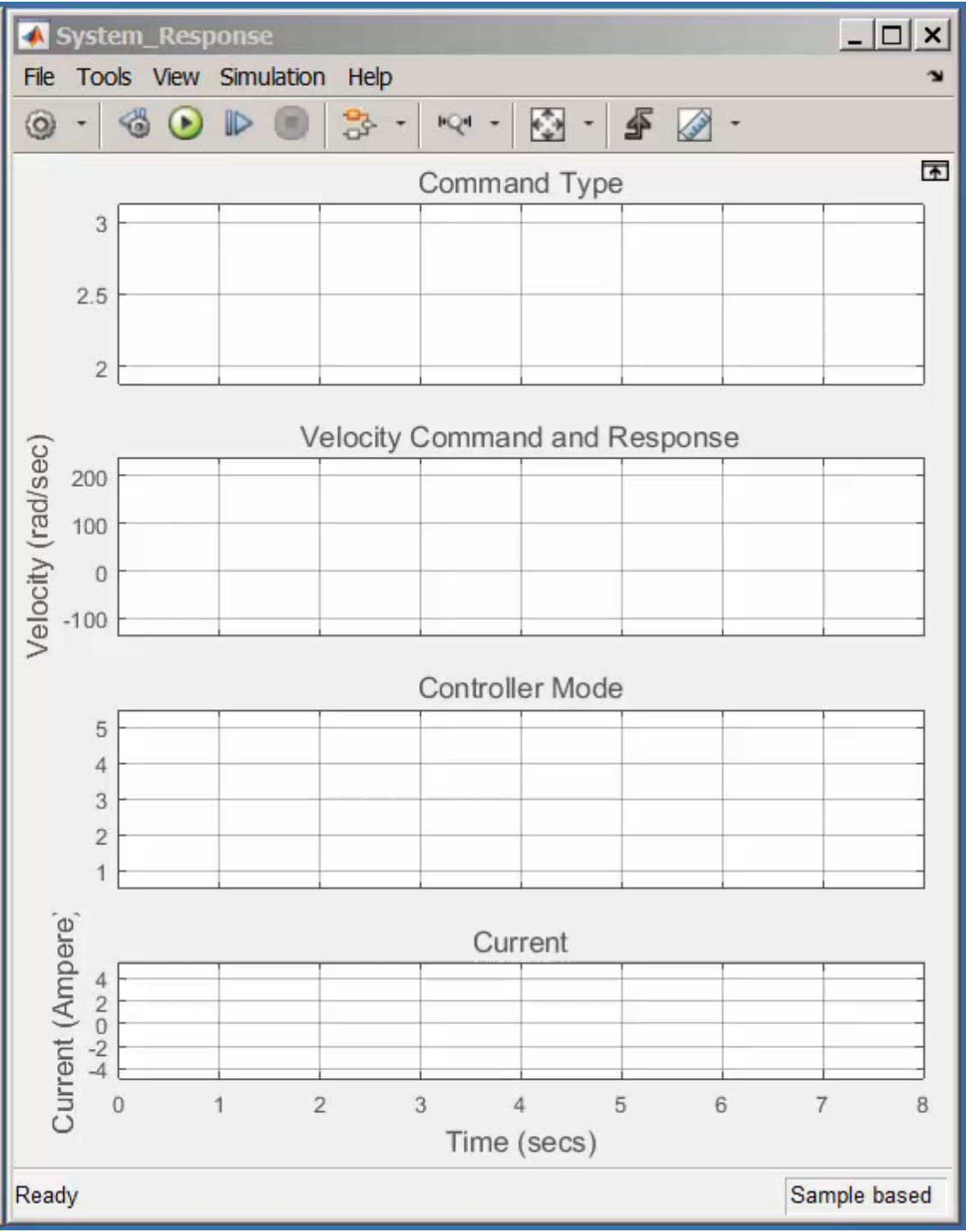
Mechanical coupler



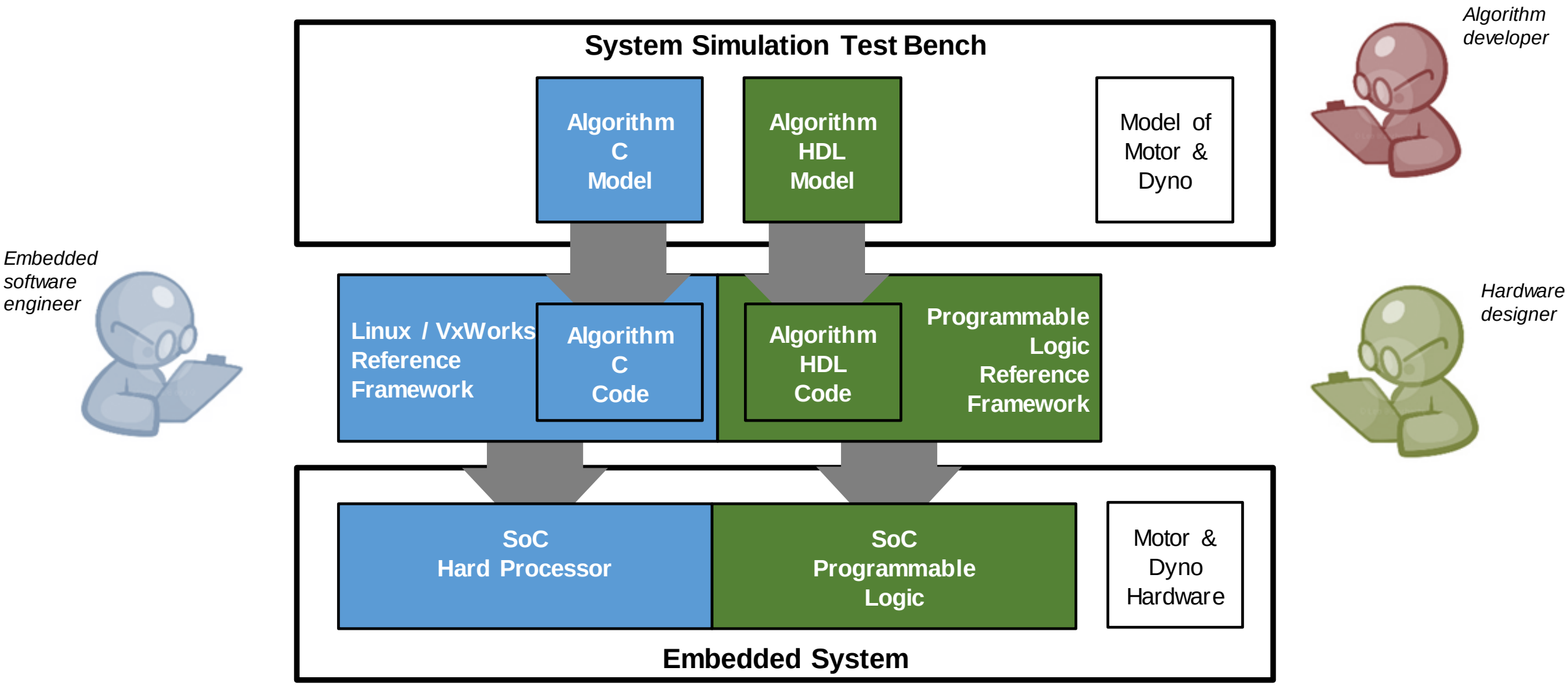
FMC module: control board + low-voltage board

Motor under test (with encoder)

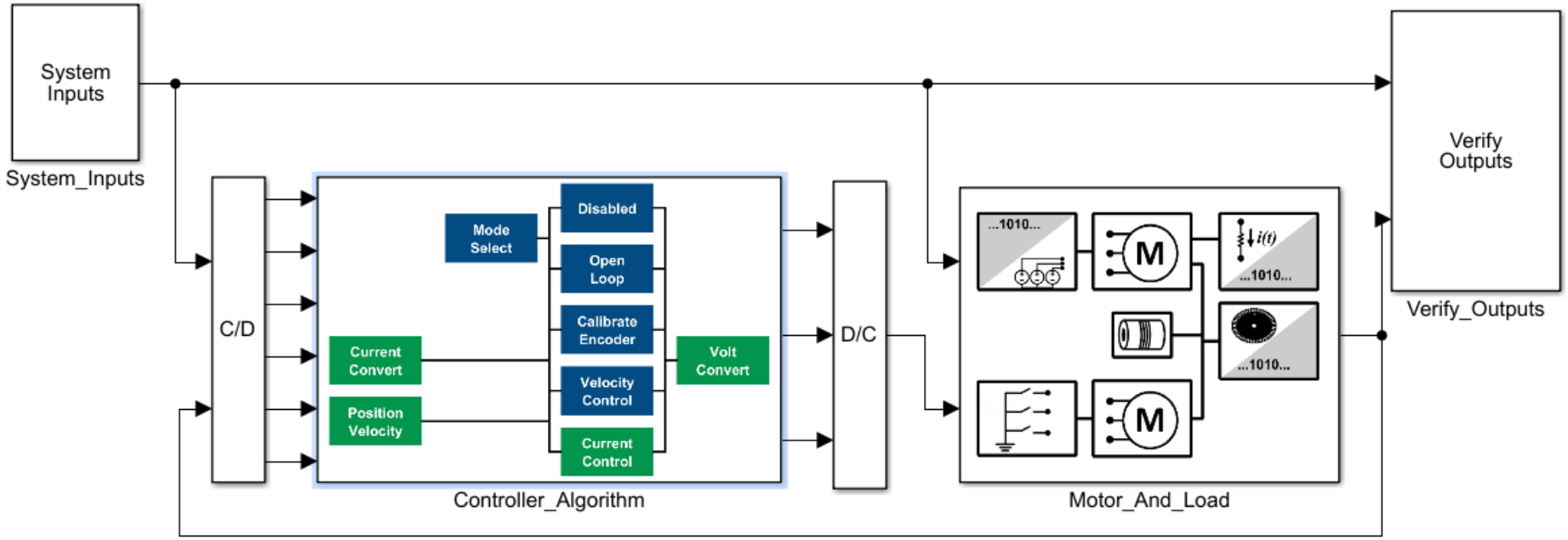

focZynqTestBench - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
focZynqTestBench
focZynqTestBench
Field-Oriented Control of Velocity
Hardware/Software Test Bench
Copyright 2015-2017 The MathWorks, Inc.
System Inputs
System_Inputs
C/D
Controller_Algorithm
D/C
Motor_And_Load
Verify Outputs
Verify_Outputs
Ready
View 1 warning 68%
VariableStepAuto


System_Response
File Tools View Simulation Help
Command Type
Velocity Command and Response
Velocity (rad/sec)
Controller Mode
Current
Current (Ampere)
Time (secs)
Ready
Sample based

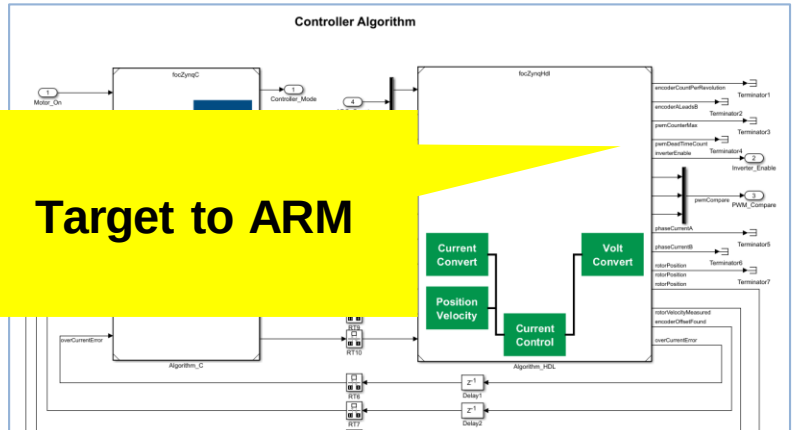
Conceptual workflow targeting SoCs



Hardware/software partitioning

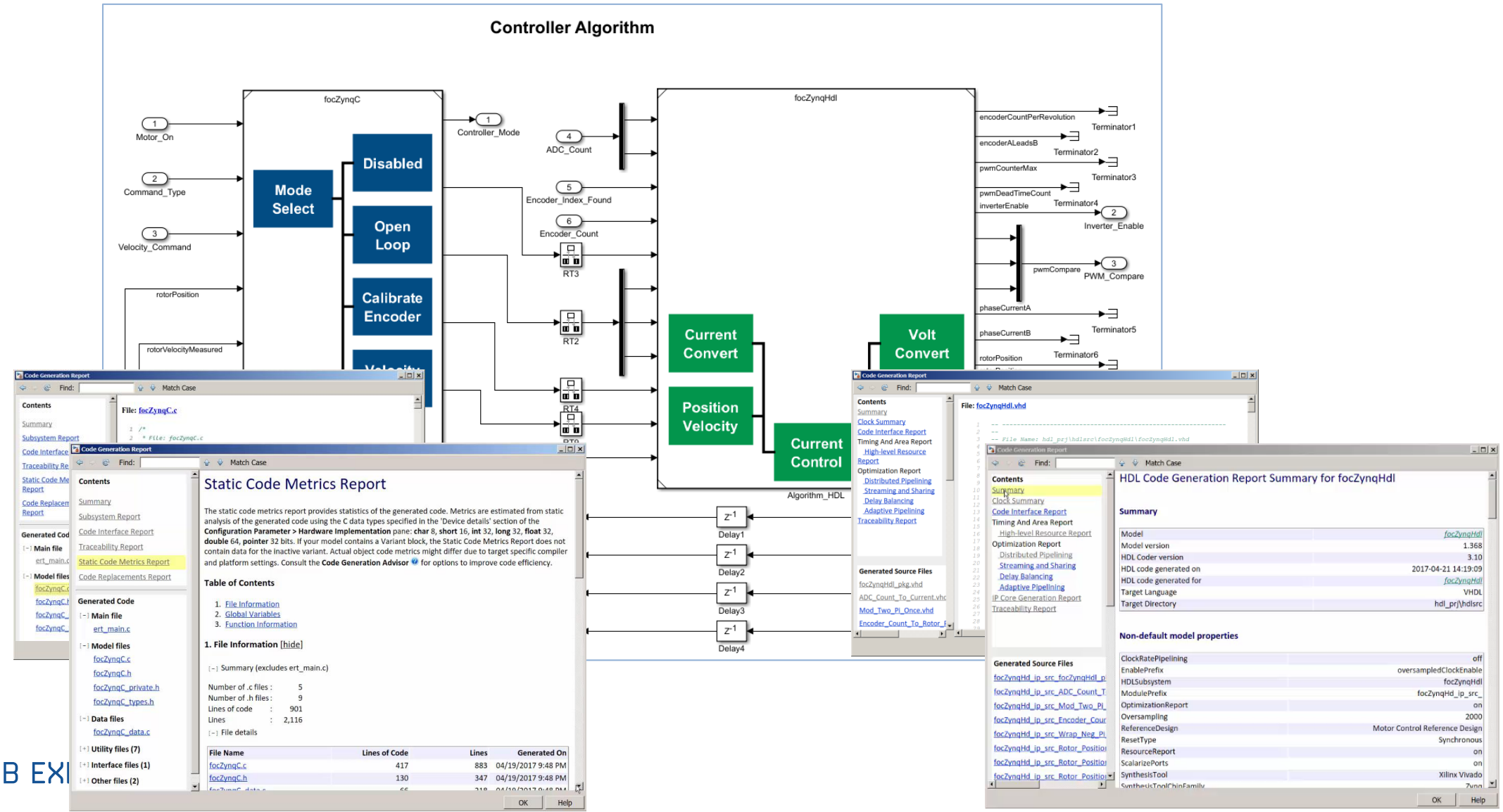


Target to Programmable Logic



Target to ARM

Code Generation



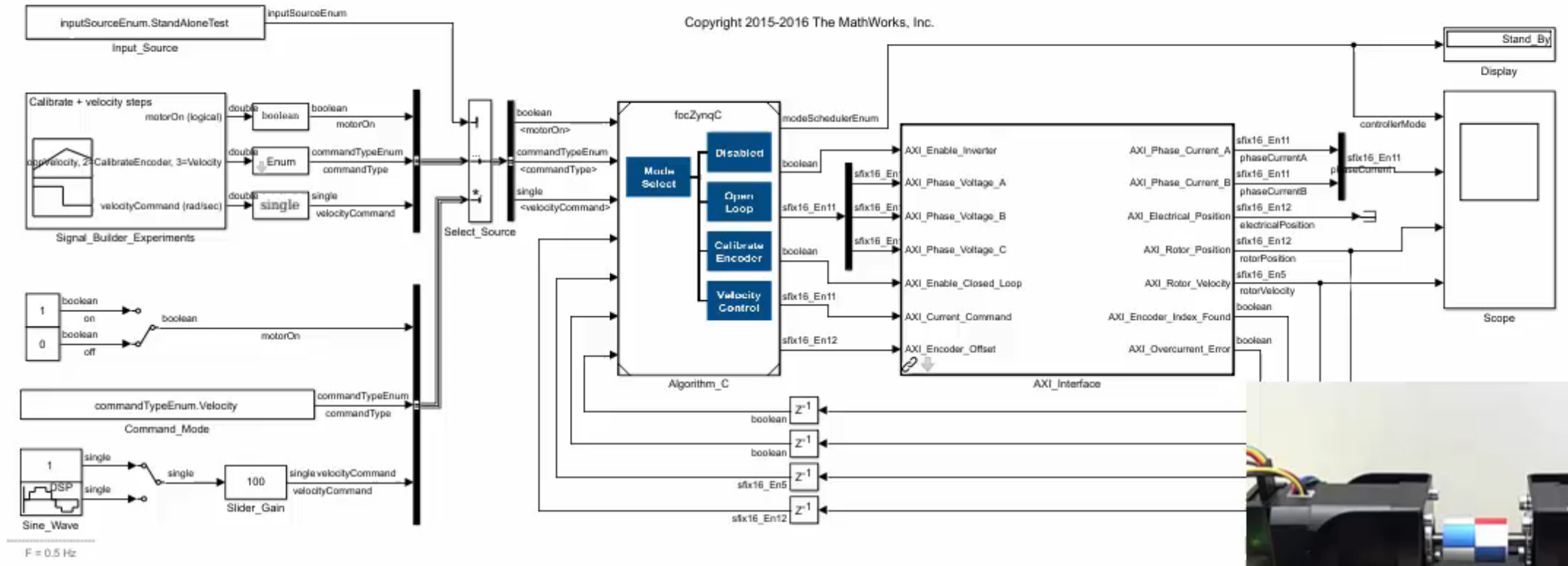
focZynqArmDeployment - Simulink

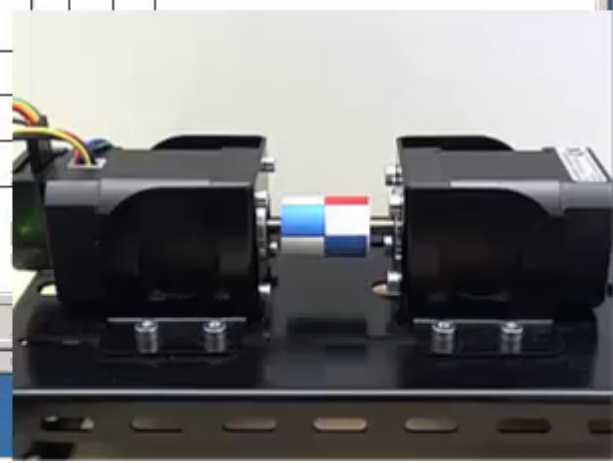
File Edit View Display Diagram Simulation Analysis Code Tools Help

focZynqArmDeployment

focZynqArmDeployment

Field-Oriented Control of Velocity
Zynq ARM Deployment for AD-FMCMOTCON2





Ready

View diagnostics 61%

3T Develops Robot Emergency Braking System with Model-Based Design

Challenge

Design and implement a robot emergency braking system with minimal hardware testing

Solution

Model-Based Design with Simulink and HDL Coder to model, verify, and implement the controller

Results

- Cleanroom time reduced from weeks to days
- Late requirement changes rapidly implemented
- Complex bug resolved in one day

A SCARA robot.

"With Simulink and HDL Coder we eliminated programming errors and automated delay balancing, pipelining, and other tedious and error-prone tasks. As a result, we were able to easily and quickly implement change requests from our customer and reduce time-to-market."

Ronald van der Meer
3T

Link to user story

Why use Model-Based Design to develop motor control applications on SoCs?

Challenges:

- Integration of software and hardware partitions of algorithm on SoC drives need for collaboration
- Validation of design specifications with limits on access to motors in labs.
- How to make design decisions that cut across system?

Model-Based Design

- ✓ Enables early validation of specifications using simulation months before hardware is available.
- ✓ Improves design team collaboration and designer productivity by using a shared design environment.
- ✓ Reduces hardware testing time by 5x by shifting design from lab to the desktop

Learn More

- Videos
  - HDL Coder: Native Floating Point
- Webinars
  - Prototyping SoC-based Motor Controllers on Intel SoCs with MATLAB and Simulink
  - How to Build Custom Motor Controllers for Zynq SoCs with MATLAB and Simulink
- Articles
  - How Modeling Helps Embedded Engineers Develop Applications for SoCs (MATLAB Digest)
  - MATLAB and Simulink Aid HW-SW Codesign of Zynq SoCs (Xcell Software Journal)
- Tutorials:
  - Define and Register Custom Board and Reference Design for SoC Workflow
  - Field-Oriented Control of a Permanent Magnet Synchronous Machine on SoCs